

Gas–oil ratio correlation (R_s) for gas condensate using genetic programming

K. A. Fattah

Received: 28 May 2013 / Accepted: 6 January 2014 / Published online: 21 January 2014
© The Author(s) 2014. This article is published with open access at Springerlink.com

Abstract A new correlation for solution gas–oil ratio (R_s) for gas condensate reservoir was developed in this paper by using genetic programming algorithm of a commercial software (Discipulus) program. Matching PVT experimental data with an equation of state model, a commercial simulator (Eclipse simulator) was used to calculate the solution gas–oil ratio (R_s) values used in this study. More than 1,800 solution gas–oil ratio (R_s) values obtained from the analysis of eight gas condensate fluid PVT laboratory reports, selected under a wide range of reservoir temperature and pressure, composition and condensate yield, were used. Comparisons of the results showed that currently published correlations of gas–oil ratio (R_s) for gas condensate gave poor estimates of its value (the average absolute error for Standing correlation was 63.48 with a standard deviation (SD) equal to 0.724, the average absolute error for Glaso correlation was 61.19 % with a SD equal to 0.688, the average absolute error for Vasques and Beggs correlation was 52.22 % with a SD equal to 0.512, the average absolute error for Marhoun correlation was 56.34 % with a SD equal to 0.519 and the average absolute error for Fattah et al. correlation was 18.6 % with a SD equal to 0.049). The proposed new correlation improved extensively the average absolute error for gas condensate fluids. The average absolute error for the new correlation

was 10.54 % with a SD equal to 0.035. Also, the hit-rate (R^2) of the new correlation was 0.9799 and the fitness variance was 0.012. The importance of the new correlation comes from depending only on readily available production data in the field and can have wide applications when representative PVT lab reports are not available.

Keywords Gas–oil ratio correlation · PVT laboratory report · Genetic programming · Gas condensate · Modified black oil simulation

Introduction

Material balance equation is a useful method of reservoir performance analysis. It is routinely used to estimate oil, and gas reserves and predict future reservoir performance. Schilthuis, in 1936, was among the first to formulate and apply material balance analysis. As time progressed, more sophisticated material balance models evolved, each striving for greater generality.

Application of two-hydrocarbon-component, zero-dimensional material balance model had been restricted to black-oil or dry-gas reservoirs. As gas condensate reservoirs exploration increases, there has been a growing need to address this limitation.

Spivak and Dixon (1973) introduced the modified black oil (MBO) simulation approach. The PVT functions for MBO simulation and material balance calculations of gas condensate are (condensate–gas ratio R_v , solution gas–oil ratio R_s , oil formation volume factor B_o , and gas formation volume factor B_g). The MBO approach assumes that stock-tank liquid component can exist in both liquid and gas phases under reservoir conditions in gas condensate reservoir.

K. A. Fattah (✉)
Petroleum and Natural Gas Engineering Department,
College of Engineering, King Saud University, P.O. 800,
Riyadh 11421, Saudi Arabia
e-mail: kelshreef@ksu.edu.sa

K. A. Fattah
Faculty of Engineering, Cairo University, Giza, Egypt

A few authors have addressed the question of how best to generate the PVT properties for gas condensate. Whitson and Tropic (1983) used laboratory constant volume depletion (CVD) data to calculate “MBO” PVT fluid properties B_o , R_s , B_g and R_v for gas condensate fluids. Coats (1985) suggested a different approach from Whitson and Tropic’s (W&T) to calculate the MBO properties for gas condensates. Walsh and Towler (1994) suggested a new simple method to compute the black-oil PVT properties of gas condensate reservoirs. Fevang et al. (2000) presented guidelines to help engineers choose between MBO and compositional approaches. Fattah et al. (2009) presented new correlations to develop MBO PVT properties when PVT fluid samples reports are not available.

Most of the methods in the literature for generating MBO PVT fluid properties (B_o , R_s , B_g and R_v) for gas condensate need a combination of lab experiments and elaborate calculation procedures.

This study involves two parts: the first part includes a comparison between the different correlations used to calculate the solution gas–oil ratio (R_s) for gas condensate to determine the most accurate one. The second part involves the development of a new correlation to calculate R_s for gas condensate reservoir using genetic algorithm methods. Validation of the new correlation is achieved through comparison between the new correlation value of R_s and R_s generated by Whitson and Tropic method from PVT lab data.

Fluid samples

Eight gas condensates (GC) samples are used in this study. The samples were obtained from reservoirs representing different locations and depth, and were selected to cover a wide range of gas condensate fluid characteristics. Some samples represent near-critical fluids as explained by McCain and Bridges (1994). Table 1 presents a description of the major properties of these eight fluid samples.

EOS models in a commercial simulator (Eclipse simulator) were used to develop an EOS model for each sample in Table 1. Tuning the EOS model that matched as best as possible the experimental results of all available PVT laboratory experiments (CCE, DL, CVD, and separator tests) was constructed. The procedure suggested by Coats and Smart (1986) to match the laboratory results was followed. For consistency, all EOS models were developed using Peng and Robinson (1976) EOS with volume shift correction (3-parameter EOS).

Approach

The developed EOS model for each sample in Table 1 was used to output MBO PVT properties (R_v , R_s , B_o , and B_g) at six different separator conditions using Whitson and Tropic (1983) procedure. The extracting data for the MBO PVT properties involves 1,836 points from the different eight gas condensate samples. The first part was to compare

Table 1 Characteristics of fluid samples

Property	GC 1	GC 2	GC 3	GC 4	GC 5	GC 6	GC 7	GC 8
Reservoir temperature (°F)	312	286	238	256	186	312	300	233
Initial reservoir pressure (psig)	14,216	NA	6,000	7,000	5,728	14,216	5,985	17,335
Initial producing gas–oil ratio (SCF/STB)	3,413	4,278	NA	4,697	5,987	8,280	6,500	6,665
Stock oil gravity (° API)	45.6	NA	NA	46.5	58.5	50.7	45.6	43
Saturation pressure (psig)	5,210	5,410	4,815	6,010	4,000	5,465	5,800	11,475
Components	Composition (mole %)							
CO ₂	2.66	4.48	0.14	0.01	0.18	2.79	6.98	0.36
N ₂	0.17	0.70	1.62	0.11	0.13	0.14	1.07	0.31
C ₁	59.96	66.24	63.06	68.93	61.72	66.73	65.25	81.23
C ₂	7.72	7.21	11.35	8.63	14.1	10.22	8.92	5.54
C ₃	6.50	4.00	6.01	5.34	8.37	5.90	4.81	2.66
iC ₄	1.93	0.84	1.37	1.15	0.98	1.88	0.85	0.62
nC ₄	3.00	1.76	1.94	2.33	3.45	2.10	1.75	1.06
iC ₅	1.64	0.74	0.84	0.93	0.91	1.37	0.65	0.47
nC ₅	1.35	0.87	0.97	0.85	1.52	0.83	0.69	0.52
C ₆	2.38	0.96	1.02	1.73	1.79	1.56	0.83	0.84
C ₇₊	12.69	12.2	11.68	9.99	6.85	6.48	8.2	6.39

between the extracted R_s and the most common R_s correlations to determine the most accurate one.

The second part involved the development of a new correlation to calculate R_s for gas condensate reservoir using genetic algorithm program.

Genetic programming

Genetic algorithms, evolution strategies and genetic programming belong to the class of probabilistic search procedures known as evolutionary algorithms that use computational models of natural evolutionary processes to develop computer-based problem-solving systems. Solutions are obtained using operations that simulate the evolution of individual structures through mechanism of reproductive variation and fitness-based selection. Due to their reported robustness in practical applications, these techniques are gaining popularity and have been used in a wide range of problem domain. The main difference between genetic programming and genetic algorithm is the representation of the solution. Genetic programming creates computer programs as solution, whereas genetic algorithm creates a string of numbers to represent the solution. Genetic programming is based on the Darwinian principle of reproduction and survival of the fittest and analogs of naturally occurring genetic operations such as crossover and mutation (Koza 1992). Genetic programming uses four steps to solve a problem (Koza 1997):

1. Generate an initial population of random compositions of the functions and terminals (input) of the problem.
2. Execute each program in the population and assign a fitness value.
3. Create a new offspring population of computer programs by copying the best programs and creating new ones by mutation and crossover.
4. Designation of the best computer program in the generation.

Solution gas–oil ratio (R_s) correlations

This part presented the comparison between the common correlations used to calculate the solution gas–oil ratio (R_s) for gas condensate in the literature. The comparison of the Vasques and Beggs correlation (1980) with the observed R_s for gas condensate result in average absolute error of 52.22 % with a SD equal to 0.512. Figure 1 presents cross-plots for R_s (Vasques and Beggs) vs. R_s (observed) for gas condensate samples. The comparison of the Standing correlation (1947) results in average absolute error of 63.48 % with a SD equal to 0.724. Figure 2 shows cross-plots for

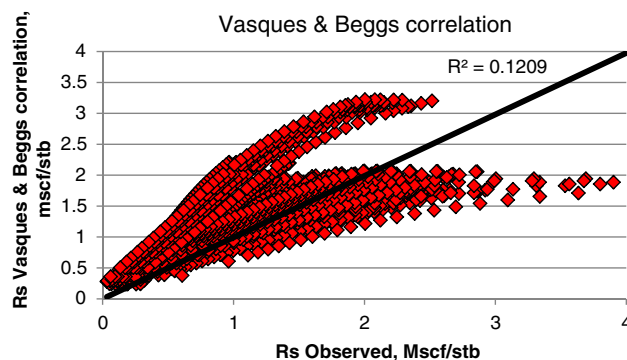


Fig. 1 R_s for gas condensate samples cross-plot for Vasques and Beggs correlation

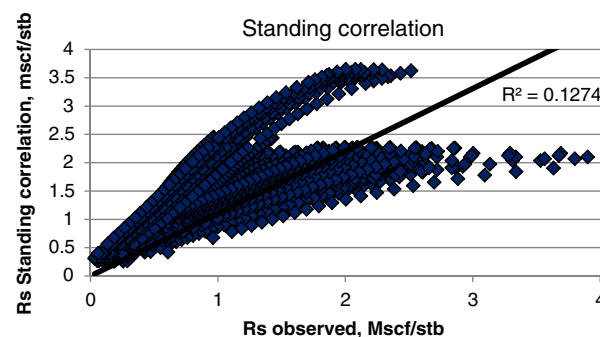


Fig. 2 R_s for gas condensate samples cross-plot for Standing correlation

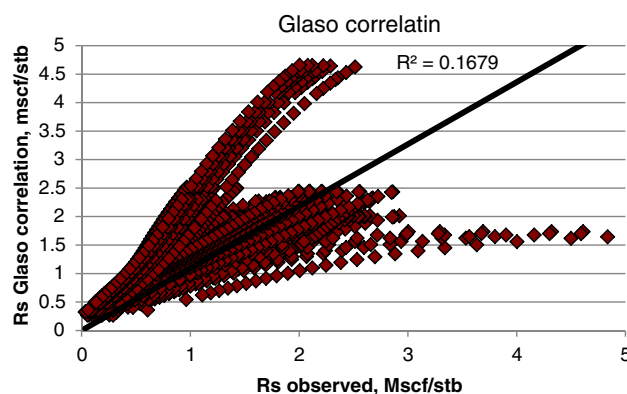


Fig. 3 R_s for gas condensate samples cross-plot for Glaso correlation

Standing correlation. The comparison of the Glaso correlation (1980) results in average absolute error of 61.19 % with a SD equal to 0.688. Figure 3 displays cross-plots for Glaso correlation. The comparison of the Marhoun correlation (1988) results in average absolute error of 56.34 % with a SD equal to 0.519. Figure 4 presents cross-plots for Glaso correlation. The comparison of the Fattah et al. correlation (2009) results in average absolute error of 18.66 % with a SD equal to 0.049, which is the best correlation in the literature. Figure 5 shows cross-plots for Fattah et al. correlation.

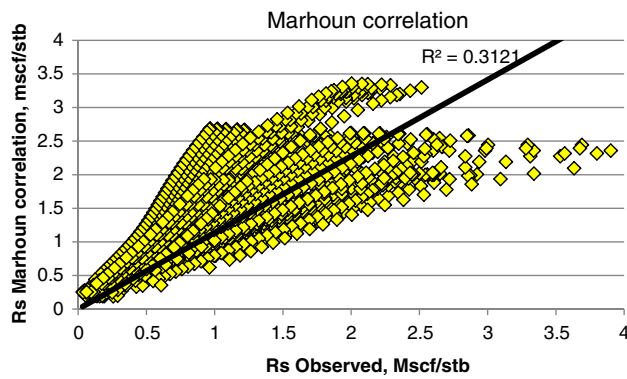


Fig. 4 R_s for gas condensate samples cross-plot for Marhoun correlation

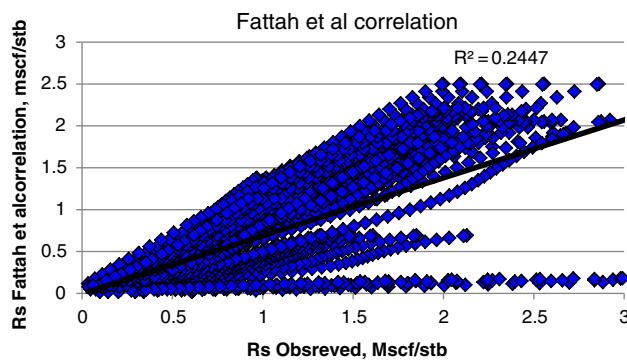


Fig. 5 R_s for gas condensate samples cross-plot for Fattah et al. correlation

Developed gas–oil ratio correlation (R_s) for gas condensate using Genetic Program

The second part in this study involved the development of the new correlation to calculate R_s for gas condensate reservoir using genetic algorithm program. A commercial Genetic Programming system called Discipulus was used to develop the new R_s correlation (Foster 2001, Francone 2004). “Discipulus is a steady state genetic programming system, using tournament selection in which two pairs of individuals compete each round for reproduction. All the usual parameters can be adjusted with Discipulus: crossover rate, mutation rate, population size, instruction set, distribution of initial program sizes, termination criteria, and parsimony pressure (fitness advantages for smaller programs).” (Foster 2001) The default settings for a Discipulus project work quite well for almost all projects. In fact, Discipulus automatically sets, randomizes, and optimizes the Genetic Programming parameters for the runs that comprise a project. For that the default setting was used in our run. The values for the default setting are: the selection method is tournament selection, the probability of mutation rate frequency is 90 %, the crossover frequency is 50 %,

and the population size (sets the number of programs in the population that Discipulus will evolve) is 500. There are two parameters that control the size of the programs evolved using Discipulus. Initial Program Size (in bytes) sets the size of the programs in the first population created by Discipulus at the start of a run (80 byte in our project). Maximum Program Size sets the maximum length of the body of an evolved program in the population (512 bytes in our project). The Genetic Programming algorithm uses a “fitness function” to determine which evolved programs survive and reproduce. The fitness function used depends on whether you present a classification problem or a regression problem to Discipulus, our problem is a regression problem. Generally speaking, the better an evolved program models your training data, the more fit it will be. Discipulus calculates the fitness of evolved programs by determining how closely the outputs of the evolved program and the target outputs in the training data match up. The closer the match, the fitter the evolved program. The two parameters used as fitness measurements are the hit-rate (R^2) of the best genetic program and the fitness variance.

The input data files for this software are classified into three semi-equal groups, “training data”, “validation data” and “applied data”. These input files include measured inputs and outputs parameters for our correlation.

The inputs parameters for our correlation are:

- Pressure (P), psi;
- Reservoir temperature (T), R° ;
- API gravity of the reservoir fluid;
- Specific gravity of surface gas (SG_g);
- Specific gravity of surface oil (SG_o).

The output is the solution gas–oil ratio, R_s .

Discipulus program gives different types of data and charts that show how the run in progress improved its performance. Discipulus creates thousands of models (programs) from given data files that allow us to predict outputs from similar inputs and for each model (program) gives us its performance [the hit-rate (R^2) and the fitness variance]. At the end of the run, we choose the best model (program) depending on its hit-rate (R^2) and fitness variance to calculate the solution gas–oil ratio; R_s .

Figure 6 shows the fitness improvement of the best genetic program for our correlation with time. The hit-rate (R^2) of the best genetic program (the new correlation), was 0.9799 and the fitness variance was 0.012. Figures 7, 8 and 9 present the match between the observed R_s and the calculated R_s for the new correlation. Each figure shows that the match between all input points of the observed R_s and the calculated R_s for the same point from the best program developed by the software, the best team (During a project, Discipulus assembles the best programs into teams. The output from all of the programs that comprise a team are

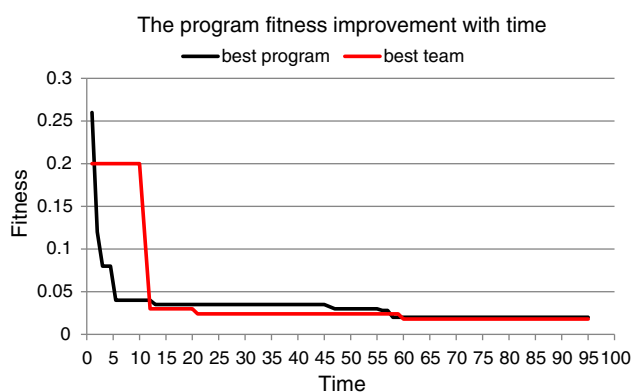


Fig. 6 The program fitness improvement with time

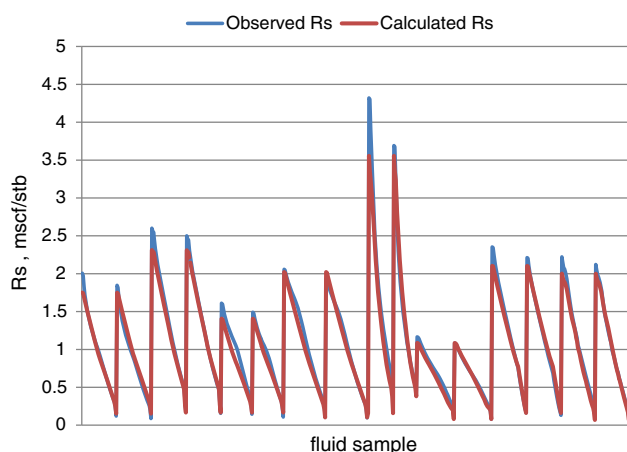


Fig. 7 The observed vs. calculated R_s data from training input data

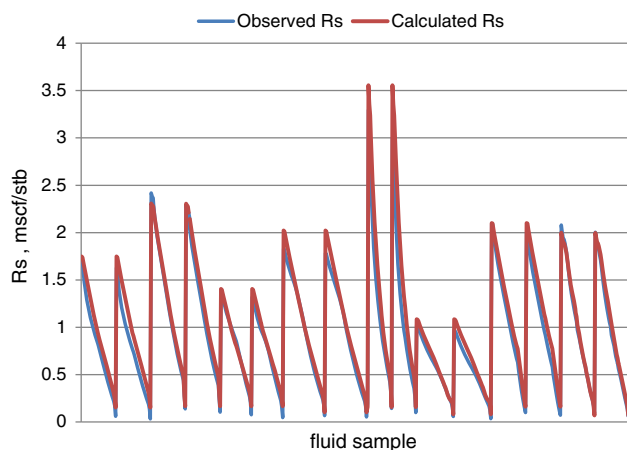


Fig. 8 The observed vs. calculated R_s data from validation input data

assembled into one collective output that is frequently better than any particular member of the team), and the selected program (almost is the best program if the default of the software is not changed). The results for our case indicated that the new correlation (best program output

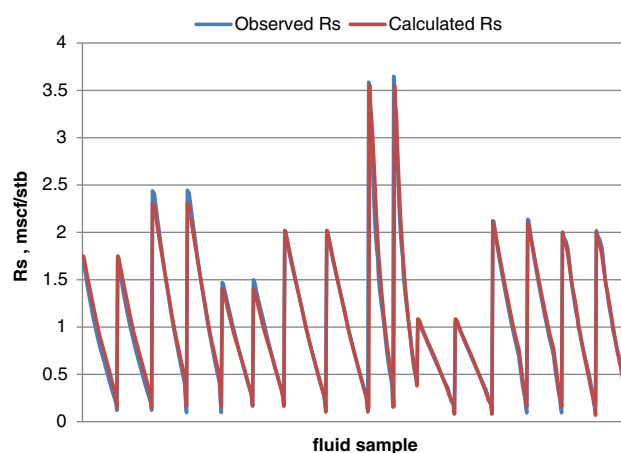


Fig. 9 The observed vs. calculated R_s data from applied input data

from the software) almost completely matches the observed R_s data.

The model outputs from the software are created as computer programs in Java, C++ code, or assembler program. For that, this correlation is a regression model correlation. So the second step to use this correlation is to run the code resulting from the genetic algorithm to get the value of the correlation. The output C++ code of the genetic model correlation from the Discipulus to calculate the new R_s correlation is given in the [Appendix](#). This code was used with C++ compiler to develop a windows interface program to calculate R_s value (Fig. 10). This code can be modified to generate a solution gas–oil ratio array for different temperature, pressure values of a given reservoir.

For more model validation, cross-plots between observed and calculated R_s were drawn (Fig. 11) and the average absolute error and the SD for the new correlation was calculated and equal to 10.54 % and 0.035, respectively. Table 2 summarizes the statistical comparison between the different correlations and the new correlation. From this table, we found that the new correlation is the best matched correlation.

The new correlation presented in this work can be used with other set of correlations to generate MBO PVT properties for material balance calculation, or reservoir simulation without the need for fluid samples or elaborate procedure for EOS calculations. The application of these correlations is of particular importance, especially when representative fluid samples are not available.

Conclusions

Based on work presented in this study, the following conclusions were made:

Fig. 10 The windows interface of the C++ program

```

A COMPUTER PROGRAM TO CALCULATE THE GAS OIL RATIO FOR GAS
CONDENSITE USING GENATIC PROGRAMING

<<<INPUT DATA>>>

Inter PRESSURE value <psia> = 1000

Inter TEMPERATURE value <R> = 689

Inter API value = 47.3255

Inter OIL S.G value = 0.79127

Inter GAS S.G value = 0.91896

<<<OUTPUT DATA>>>

The GAS OIL RATIO ARE <Mscf/STB> = 0.159353

```

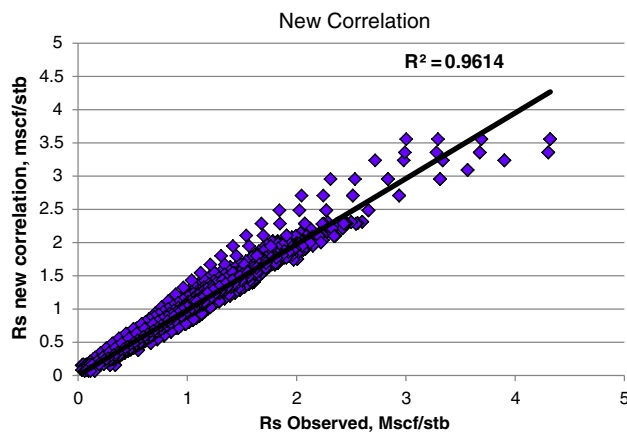


Fig. 11 R_s for gas condensate samples cross-plot for new correlation

1. The comparison of the literature gas–oil ratio correlations for gas condensate shows that Fattah et al. correlation was the best correlation to calculate gas–oil ratio R_s with average absolute error of 18.66 % and a SD equal to 0.049.
2. New R_s correlation was developed for gas–oil ratio R_s of gas condensates. The Discipulus software, a commercial Genetic Programming system, was used to develop the new correlation program. It is based on the concept of genetic algorithm.
3. The hit-rate (R^2) of the new correlation was 0.9799 and the fitness variance for the new correlation was 0.012.

4. Comparison of the new correlation with the observed gas–oil ratio R_s result in average absolute error of 10.54 % with a SD equal to 0.035.

Open Access This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

Appendix

This appendix gives the output C++ code of the best genetic program from the Discipulus to calculate the new R_s correlation. This program is a sequential type program.

```

*****
*****
#define TRUNC(x)((x)>=0) ? floor(x) : ceil(x)
#define C_FPREM (_finite(f[0]/f[1]) ? f[0]-
(TRUNC(f[0]/f[1])*f[1]) : f[0]/f[1])
#define C_F2XM1 (((fabs(f[0])<=1) && (!_is-
nan(f[0]))) ? (pow(2,f[0])-1) : ((!_finite(f[0]) && !_is-
nan(f[0]) && (f[0] < 0)) ? -1 : f[0]))

float DiscipulusCFunction(float v[])
{
    long double f[8];
    long double tmp=0;

```

Table 2 Statistical comparison of all correlations using observed data

Vas and Beggs correlation		Standing correlation		Glaso correlation		Marhoun correlation		Khaled correlation		New correlation	
AAE	SD	AAE	SD	AAE	SD	AAE	SD	AAE	SD	AAE	SD
52.2 %	0.51	63.4 %	0.73	61.2 %	0.69	56.3 %	0.52	18.6 %	0.05	10.5 %	0.035

```

int cflag=0;

f[0]=f[1]=f[2]=f[3]=f[4]=f[5]=f[6]=f[7]=0;
L0: f[0]/=-1.238061666488648f;
L1: f[0]+=0.9177978038787842f;
L2: f[0]=-f[0];
L3: tmp=f[3]; f[3]=f[0]; f[0]=tmp;
L4: f[0]/=f[3];
L5: f[0]=-f[2];
L6: cflag=(f[0]<f[2]);
L7: f[2]*=f[0];
L8: cflag=(f[0]<f[3]);
L9: tmp=f[0]; f[0]=f[0]; f[0]=tmp;
L10: cflag=(f[0]<f[3]);
L11: f[0]=cos(f[0]);
L12: f[0]-=v[2];
L13: f[0]=fabs(f[0]);
L14: f[0]*=-0.7297487258911133f;
L15: f[0]=sin(f[0]);
L16: f[0]=-f[0];
L17: f[3]-=f[0];
L18: f[0]=sin(f[0]);
L19: f[0]=-1.174947738647461f;
L20: if (cflag) f[0]=f[3];
L21: f[3]-=f[0];
L22: f[0]-=f[0];
L23: f[0]*=pow(2,TRUNC(f[1]));
L24: f[0]=cos(f[0]);
L25: f[0]+=f[3];
L26: f[0]=sin(f[0]);
L27: if (!cflag) f[0]=f[0];
L28: f[0]=fabs(f[0]);
L29: f[0]*=pow(2,TRUNC(f[1]));
L30: f[0]=-f[0];
L31: f[2]+=f[0];
L32: f[0]-=f[2];
L33: f[0]=cos(f[0]);
L34: f[0]*=v[2];
L35: f[0]=sin(f[0]);
L36: f[0]=-f[0];
L37: cflag=(f[0]<f[1]);
L38: f[2]/=f[0];
L39: f[0]+=f[0];
L40: f[3]-=f[0];
L41: f[3]-=f[0];
L42: f[0]=sin(f[0]);
L43: f[1]+=f[0];
L44: f[0]+=1.258495330810547f;
L45: f[2]*=f[0];
L46: f[0]=cos(f[0]);
L47: f[0]+=-1.360518217086792f;
L48: f[0]*=v[2];
L49: f[0]=sin(f[0]);

L50: f[0]=-f[0];
L51: f[0]/=0.1387641429901123f;
L52: f[0]+=1.77857518196106f;
L53: f[3]=-f[0];
L54: f[0]=sin(f[0]);
L55: f[0]+=f[0];
L56: f[0]=fabs(f[0]);
L57: f[3]=-f[0];
L58: f[0]*=f[3];
L59: f[0]=-f[1];
L60: f[2]/=f[0];
L61: tmp=f[2]; f[2]=f[0]; f[0]=tmp;
L62: f[0]/=v[4];
L63: f[2]+=f[0];
L64: f[0]*=pow(2,TRUNC(f[1]));
L65: if (!cflag) f[0]=f[3];
L66: f[0]/=f[1];
L67: f[0]/=f[0];
L68: f[0]*=-1.174947738647461f;
L69: f[0]=fabs(f[0]);
L70: f[0]=sin(f[0]);
L71: f[3]-=f[0];
L72: f[0]=sin(f[0]);
L73: f[0]+=f[0];
L74: f[3]-=f[0];
L75: if (!cflag) f[0]=f[3];
L76: f[0]=cos(f[0]);
L77: f[0]*=v[0];
L78: f[0]+=f[1];
L79: f[0]*=f[1];
L80: f[0]-=v[0];
L81: f[0]=fabs(f[0]);
L82: f[0]=sqrt(f[0]);
L83: f[0]-=f[3];
L84: f[0]/=v[2];
L85: f[3]*=f[0];
L86: if (!cflag) f[0]=f[1];
L87: f[0]*=f[0];
L88: cflag=(f[0]<f[3]);
L89: f[0]/=f[3];
L90: f[0]=sin(f[0]);
L91: f[0]=-f[0];
L92: f[3]-=f[0];
L93: f[1]+=f[0];
L94: tmp=f[1]; f[1]=f[0]; f[0]=tmp;
L95: f[3]-=f[0];
L96: f[0]/=f[2];
L97: f[0]=-f[0];
L98: cflag=(f[0]<f[1]);
L99: if (!cflag) f[0]=f[3];
L100: f[0]+=f[2];
L101: f[0]=cos(f[0]);
L102: f[0]*=v[1];

```

```

L103: f[0]+=f[0];
L104: f[0]=fabs(f[0]);
L105: f[0]=-v[0];
L106: f[0]=-f[2];
L107: f[0]=fabs(f[0]);
L108: f[0]=sqrt(f[0]);
L109: f[0]=-f[3];
L110: f[0]/=v[2];
L111: f[3]*=f[0];
L112: f[0]=sqrt(f[0]);
L113: f[2]=-f[0];
L114: tmp=f[1]; f[1]=f[0]; f[0]=tmp;
L115: f[2]+=f[0];
L116: f[3]=-f[0];
L117: f[2]+=f[0];
L118: if (cflag) f[0]=f[2];
L119: f[0]*=f[3];
L120: f[0]=-f[3];
L121: f[0]*=pow(2,TRUNC(f[1]));
L122: f[0]=-v[0];
L123: f[0]=-f[3];
L124: f[0]=-f[0];
L125: f[0]=sqrt(f[0]);
L126: f[0]=-f[3];
L127: f[0]/=v[2];
L128: f[3]*=f[0];
L129: f[0]=-f[0];
L130: f[0]=sin(f[0]);
L131: tmp=f[1]; f[1]=f[0]; f[0]=tmp;
L132: f[0]=-0.9765300750732422f;
L133: f[0]=sin(f[0]);
L134: f[0]=-f[0];
L135: f[0]*=f[1];
L136: f[3]/=f[0];
L137: f[0]*=pow(2,TRUNC(f[1]));
L138: f[0]=-f[0];
L139: f[0]+=f[1];
L140: if (cflag) f[0]=f[1];
L141: tmp=f[0]; f[0]=f[0]; f[0]=tmp;
L142: f[3]=-f[0];
L143: cflag=(f[0]<f[2]);
L144: f[3]=-f[0];
L145: f[0]=cos(f[0]);
L146: f[0]*=v[1];
L147: tmp=f[0]; f[0]=f[0]; f[0]=tmp;
L148: f[0]*=f[0];
L149: f[0]/=f[2];
L150: f[0]=sin(f[0]);
L151: f[0]+=f[0];
L152: f[0]/=f[1];
L153: f[0]*=f[0];
L154: cflag=(f[0]<f[2]);
L155: f[0]*=f[0];

```

```

L156: f[0]=sin(f[0]);
L157: cflag=(f[0]<f[3]);
L158: f[0]=sin(f[0]);
L159: if (cflag) f[0]=f[1];
L160: f[0]/=v[4];
L161: f[0]*=pow(2,TRUNC(f[1]));
L162: if (!cflag) f[0]=f[1];
L163: f[3]=-f[0];
L164: f[0]=sin(f[0]);
L165: f[1]+=f[0];
L166: f[3]=-f[0];
L167: f[0]=sin(f[0]);
L168: f[0]+=f[1];
L169: f[3]=-f[0];
L170: f[0]=cos(f[0]);
L171: f[0]*=v[2];
L172: f[0]+=1.366016626358032f;
L173: if (!cflag) f[0]=f[0];
L174: f[0]*=f[0];
L175: f[0]+=-v[1];
L176: f[0]=-v[0];
L177: f[0]=fabs(f[0]);
L178: f[0]=sqrt(f[0]);
L179: f[0]=-f[3];
L180: f[0]/=v[2];
L181:
if (!_finite(f[0])) f[0]=0;
return f[0];
}
//This program was evolved with Discipulus(tm).

```

References

- Coats KH (1985) Simulation of gas condensate reservoir performance. Paper SPE 10512, published at the JPT, Oct 1985, pp 1870–1886
- Coats KH, Smart GT (1986) Application of a regression-based EOS PVT program to laboratory data. SPERE, May 1986, pp 277–299
- ECLIPSE suite of programs, Schlumberger, 2005
- Fattah KA, El-Banbi AH, Sayyoub MH (2009) New correlations calculate volatile oil, gas condensate PVT properties. Oil Gas J
- Fevang O, Singh K, Whitson CH (2000) Guidelines for choosing compositional and black-oil models for volatile oil and gas-condensate reservoirs. Paper SPE 63087 presented at the 2000 SPE annual technical conference and exhibition, Dallas, TX, 1–4 Oct 2000
- Foster JA (2001) Review: Discipulus: a commercial genetic programming system. Register Machine Learning Technologies
- Francone FD (2004) Discipulus, owner's manual. Machine Learning Technologies
- Glaso O (1980) Generalized pressure–volume–temperature correlations. JPT
- Koza JR (1992) Genetic programming. In: On the Programming of Computers by Means of natural Selection, The MIT Press, Cambridge, MA

- Koza JR (1997) Genetic programming. Encyclopedia of Computer Science and Technology, 18 Aug 1997
- Marhoun MA (1988) PVT correlation for middle east crude oils. JPT
- McCain WD Jr, Bridges B (1994) Volatile oils and retrograde gases-what's the difference? Petrol Eng Inter 66(1):45-46
- Peng DY, Robinson DB (1976) A new two-constant equation of state. Ind Eng Chem Fund 15:59
- Schilthuis RJ (1936) Active oil and reservoir energy. Trans AIME 148:33–52
- Spivak VJ, Dixon TN (1973) Simulation of gas condensate reservoir. Paper SPE 4271 presented at the 3rd numerical simulation of reservoir performance symposium, Houston, 10–12 Jan 1973
- Standing MB (1947) A pressure-volume-temperature correlation for mixture of California oils and gases. Drilling and production practice. API, New York, pp 275–287
- Vasques M, Beggs (1980) HD correlation for fluid physical property prediction. JPT
- Walsh MP, Towler BF (1994) Method computes PVT properties for gas condensate. OGJ, pp 83–86
- Whitson CH, Trop SB (1983) Evaluating constant-volume depletion data. Paper SPE 10067, SPE, Richardson